Remarks

Upon entry of the foregoing preliminary amendment, claims 26-28, 30-43, 45-46, 48-50, 52-55 and 57-74 are pending in the application. Claims 29, 44, 47 and 56 are sought to be cancelled without prejudice to or disclaimer of the subject matter therein. Claims 1-25 and 51 were previously canceled. New claims 71-74 are sought to be added. These changes are believed to introduce no new matter, and their entry is respectfully requested.

Prior to the filing of this RCE, in the Final Office Action dated August 11, 2006, claims 26, 32, 57, 66, 69, 70 stand rejected under 35 U.S.C. § 112, first paragraph. Claims 26 and 30 stand rejected under 35 USC § 102(e) as being allegedly anticipated by Utsunomiya et al., U.S. Patent No. 6,101,558. Claims 33, 36-45, 48-50, 52-57, 59-61, 63-70 stand rejected under 35 USC § 102(e) as being allegedly anticipated by Jennings et al., U.S. Patent No. 6,760,763. Claims 27-28 stand rejected under 35 USC § 103(a) as being allegedly anticipated by Utsunomiya et al. in view of Bell et al., U.S. Patent No. 6,052,380. Claims 29, 31-32 stand rejected under 35 USC § 103(a) as being allegedly unpatentable over Utsunomiya et al. in view of Jennings et al. Claims 34-35, 46-47 and 58 stand rejected under 35 USC § 103(a) as being allegedly unpatentable over Utsunomiya et al. in view of Bell et al.

Rejections and objections under 35 U.S.C. § 112, first paragraph

A number of claims have been rejected based on § 112, first paragraph. The particulars of the § 112, first paragraph rejections dealt with the file pieces being of equal size. In the

interest of advancing the prosecution on this case, the independent claims have been amended to emphasize a different aspect of the invention, and this aspect canceled from the claims.

Rejections Based on Utsunomiya et al.

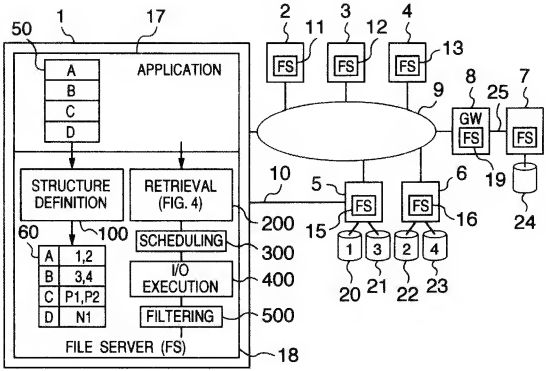
A number of claims stand rejected based on Utsunomiya in combination with Jennings. In view of the amendments to the independent claims, Applicants believe that it makes more sense to address the § 102 and § 103 rejections together. Applicants further note that the independent claims now incorporate the language some of the dependent claims (for example, dependent claim 29), which was rejected based on a combination of Utsunomiya and Jennings.

First, the amended claims all now recite

> wherein a server belonging to more than one group acts as a boundary server, and
>
> wherein boundary servers are used to transfer pieces of the file to servers of groups other than a group to which a client has connected

This topological aspect is discussed, for example, at pages 13-16 of the specification, as well as in the figures of the present application. These aspects are not disclosed anywhere in Utsunomiya – Utsunomiya has an entirely different topology, as is evident from its FIG. 1:

FIG. 1



Second, the amended claims recite that a file is divided into N pieces, the pieces stored on **N** servers, such that a file can be reconstructed from **any** K out of the N pieces. Applicants note that this is an important distinction from the cited references, since this is emphatically not true of either Utsunomiya or Jennings.

Utsunomiya is directed to a system for high speed storage and file access – as will be seen from the description below (taken from the Utsunomiya reference), Utsunomiya divides its file into regions (labeled A, B, C, D) and those regions are then stored on different disk drives. In essence, this is a RAID concept in a more "distributed" form, see the following passage from Utsunomiya:

FIG. 1 shows the structure of a high speed file system according to the invention. Computers 1 to 6, and 8 are interconnected via a network 9. The computers have file servers (FS) 11 to 13, 15, 16, and 19. The computers 1 to 4 and 8 connect the network 9 and a network 25 as their input/output (I/O) devices. The computers 1 and 5 connect a network 10 in addition to the network 9. The computers 5 and 6 connect disk devices 20 to 23 in addition to the networks 9 and 10, as their I/O devices. The disk devices 20 to 23 are discriminated by their device numbers "1", "3", "2" and "4", respectively. The computer 8 functions as a gateway to the network 25. Another computer 7 is connected to the network 25, and connects a disk device 24. **In order to access a file in the disk devices 20 to 24, a user application program 17 installed on the computer 1 instructs a file server 18 of the computer 1 to define a file structure (to be later described) and issue a file I/O request**. If the application programs of the computers 2 to 4 are executed in parallel with the application program of the computer 1, the file servers 11 to 13 of the computers 2 to 4 operate in a similar manner to the file server 18 of the computer 1. **The file servers 15, 16, and 19 on the computers 5 and 6 connecting the disk devices 20 to 23 and on the computer 8 connecting the external network 25, receive requests from the file servers of the other computers 1 to 4 and perform actual disk I/O processes to transmit the results back to the requesting computers 1 to 4**. …..

In Utsunomiya, to reconstruct a file, **all** of the regions (and the sub-regions of each region) must be recovered and put back together into a file – in Utsunomiya, if even one of the sub-regions were not accessible, Utsunomiya would generate a file access error. This provides an important distinction over Utsunomiya.

Applicants further note that this is what the language "functionally equivalent" in the specification and the claims refers to – that **any** K out of the N pieces can be used to reconstruct the file, and no one file piece is uniquely necessary for full file recovery. This is emphatically not true of Utsunomiya.

Third, the amended claims recite that N pieces are stored on N servers – Utsunomiya shows that a single server can have multiple disk drives, see, e.g., elements 20-23 in FIG. 1, and

therefore, multiple pieces of the same file can be stored on the same server – in Utsunomiya the file pieces are not stored on N servers, but on **fewer** than N servers.

Fourth, the Office Action referred to FIG. 5 of Utsunomiya and the associated discussion in Utsunomiya as disclosing a list of servers. This is incorrect. As may be seen, again, from the discussion in Utsunomiya, and FIG. 5 itself reproduced below, **FIG. 5 is a list of file regions** – labeled A, B, C, D – **it is not a list of servers**:

> **Upon reception of the file structure definition script, the file server interprets it to form the file structure table 60. FIG. 4** shows the file structure defined at the file structure defining step. In this structure, portions 602, 603 and 604 will be later described. In the structure, an upper row 600 indicates the names of the regions defined by the script, and a lower row 601 indicates that each region is distributed to what disk device. **For example, the front half of the region A is assigned to the disk device 20, whereas the back half thereof is assigned to the disk device 22. Therefore, an access to the front half of the region A is performed by always accessing the disk device 20.** FIG. 5 shows the file structure table 60. The table is constituted of, sequentially from the left column, the name of each region, a start offset 61 of each portion of the region, a length (byte number) 62 of each portion, a device number 63 of an allocated disk device, and other attributes 64. The start offset of the first portion of region A is expressed by a relative byte address (RBA) as referenced to the start address "0" of this file. Therefore, the offset and length of each of the two sub-regions of the region A can be determined relative to the whole of the file. For the region A, data of L1 bytes from the start of the file is stored in the disk device 20, and data of L2 bytes from OFT1 (=L1) is stored in the disk device 22. For the region C, since two access paths P1 and P2 are designated, these access paths are written in the other attribute column. For the region D, since this region is accessed via the other network N1 25 by using the NFS protocol, this protocol name is written in the other attribute column. **The file server 18 stores information of the file structure table 60 formed at the file structure definition step, in some disk devices such as the disk devices 20 and 21.** Referring to the disk device numbers 63, the disk device 20 is allocated to the region A and thereafter to the region C, so that data of the length L1 and data of the length L5 form a continuous storage field in the physical device 20.

FIG. 5

60	61	62	63	64
A	0	L1	1	
	OFT1	L2	2	
B	OFT2	L3	3	
	OFT3	L4	4	
C	OFT4	L5	1	P1, P2
	OFT5	L6	3	
D	OFT6	L7	N1	NFS

In fact, two regions can be stored on two disks such that those two separate physical disks are associated with a single file server, see FIG. 1, for example, elements 20-23. In this case, two separate regions can be stored on disk drives associated with the same server. This is clearly different from what is claimed, where each of the pieces of the file is stored on a separate server.

In reality, even a brief examination of the figure (with "OFT" – offsets, etc.) makes it clear that the discussion in the Office Action of FIG. 5 as disclosing a list of servers is incorrect. Reconsideration of the rejections based on Utsunomiya is respectfully requested.

Rejections Based on Jennings et al.

Addressing Jennings, Jennings fails to teach or suggest the claimed topological aspects of server organization, quoted earlier (see amended independent claims):

> wherein a server belonging to more than one group acts as a boundary server, and
>
> wherein boundary servers are used to transfer pieces of the file to servers of groups other than a group to which a client has connected

The Jennings server mirror scheme's topology has nothing in common with this aspect – this is evident even from a brief inspection of the figures of Jennings.

Furthemore, Applicants respectfully submit that the Office Action is misinterpreting what Jennings discloses. Jennings is directed to a scheme where mirror servers are used to store copies of the same file, such that any of a number of servers can deliver the same file. Jennings **never divides files into pieces**, files are always manipulated, stored and transferred as whole, indivisible entities. The Office Action refers to column 4, lines 15-40 and column 1, lines 40-58, as allegedly disclosing the aspect of dividing a file into N pieces, with K pieces being sufficient to recover the file. Respectfully, this is incorrect. Those passages are reproduced below, for the Examiner's convenience:

> An example of a process flow diagram for the monitor process to redistribute files to a set of servers is shown in FIG. 3. The process starts at 301. A timed or triggered event 303 begins the process whereby a configuration description for each server is obtained 305. This configuration includes a description of the server capacities. It is assumed that the cluster includes N servers. For example, the capacity can be given as a weighted value compared to the weighted values of the other N-1 servers in a server cluster. The server log files are obtained in step 307. **File sets are created in 309 by finding within the log files, files that are in close proximity and closely related. Close proximity files are those which are generally requested anytime some other file is requested. Using the example of a web site, an example of closely related files would correspond to URLs, wherein a main HTML page and all of the images that are referenced within that page are closely related.** ....
>
> .... **If host address fields of two or more files match within the time window, these files are grouped together and are considered to be closely related.** In the case of a web site, a further test is sometimes performed. This test looks at the protocol fields. ....

Respectfully, nothing in those passages suggests anything about dividing a file into pieces – in fact, Jennings, in numerous places, refers to **distributing files**, as is expected for a

server file mirroring scheme. For example, the following passage at col. 7, lines 15-34 is instructive:

> Still another embodiment of the invention provides a method for **restructuring content at a web site having a plurality of servers**. The method includes the steps of: obtaining a log file from each of the servers; analyzing each entry in each log file for a URL relation; and merging URLs into URL groups based on a relation rule. In some embodiments, the method further includes the steps of making available data corresponding to each file in a URL group at the corresponding server.
>
> Still another embodiment of the invention provides an apparatus for restructuring content of `N` servers in a servers cluster. **Each server has at least one file.** The apparatus includes: means for obtaining a log file from each of the servers; means for analyzing each entry in each log file for a file relation; and means for merging files into `N` file groups, wherein each file group satisfies a common file relation. In some embodiments, the means for merging files includes a means for forming file groups based on a configuration of each particular server.

The above passage further confirms that Jennings does not operate on file pieces – only on files, as would be standard in a server mirror system. FIG. 3, reproduced below, is also quite clear that Jennings works with files and file sets, not with file pieces, see particularly 307-309 in Jennings' FIG. 3 below:

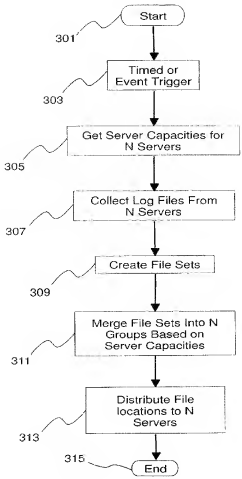


Fig. 3

The corresponding description in Jennings (FIG. 3 "shows an example of an algorithm to allocate files to each server within a group of servers in accordance with the present invention") further confirms this.

Furthermore, the Office Action referred to column 8, lines 32-38, as allegedly being relevant to this aspect. Respectfully, this is incorrect. This passage is reproduced below:

> dividing all URLs into `N` URL groups, by forming URL groups, wherein each URL group is based upon a configuration of each particular said server; providing each server with contents of the corresponding subset of URLs and activating said subset of URLs in each particular server, and updating URL links in each of the servers in accordance with said URL relocation, if relocated.

This passage, like the rest of Jennings, deals with server mirroring - nothing in this passage suggests that Jennings divides its file into pieces – all that this passage discloses is that the same file can be stored on multiple mirror servers, and the corresponding URL scheme to point to those files – however, it is the mirrored **files** that are identical, **not the pieces of the files**. In sum, Applicants respectfully submit that the Office Action misinterprets the teaching of Jennings, which has no relevance to the claimed invention. Reconsideration is respectfully requested.

New Claims 71-74

Additional topological aspects of server organization have also been added in claims 71-74. Support for the language of these new claims 71-74 may be found, for example, at pages 13-16, as well as in the figures of the present application. Applicants respectfully submit that none of these topological aspects are disclosed in any of the cited references, singly or in combination.

Conclusion

Prompt and favorable consideration of this Preliminary Amendment is respectfully requested. If the Examiner believes, for any reason, that personal communication will expedite prosecution of this application, the Examiner is invited to telephone the undersigned at the number provided.

Respectfully submitted,

BARDMESSER LAW GROUP

/GB/

George S. Bardmesser
Attorney for Applicants
Registration No. 44,020

Date: September 7, 2006

910 17th Street, N.W.
Suite 800
Washington, D.C. 20006
(202) 293-1191